LEARNING WITH SCRATCH

What do students **learn** as they create interactive stories, animations, games, music, and art with Scratch?

For one thing, they learn **mathematical and computational ideas** that are built into the Scratch experience. As students create programs in Scratch, they learn core computational concepts such as iteration and conditionals. They also gain an understanding of important mathematical concepts such as coordinates, variables, and random numbers.

Significantly, students learn these concepts in a **meaningful** and **motivating** context. When students learn about variables in traditional algebra classes, they usually feel little personal connection to the concept. But when they learn about variables in the context of Scratch, they can use variables immediately in very meaningful ways: to control the speed of an animation, or to keep track of the score in a game they are creating.

As students work on Scratch projects, they also learn about the **process of design**. Typically, a student will start with an idea, create a working prototype, experiment with it, debug it when things go wrong, get feedback from others, then revise and redesign it. It's a continuous spiral: get an idea, create a project, which leads to new ideas, which lead to new projects, and on and on.

This project-design process combines many of the **21st century learning skills** that will be critical to success in the future: thinking creatively, communicating clearly, analyzing systematically, collaborating effectively, designing iteratively, learning continuously.
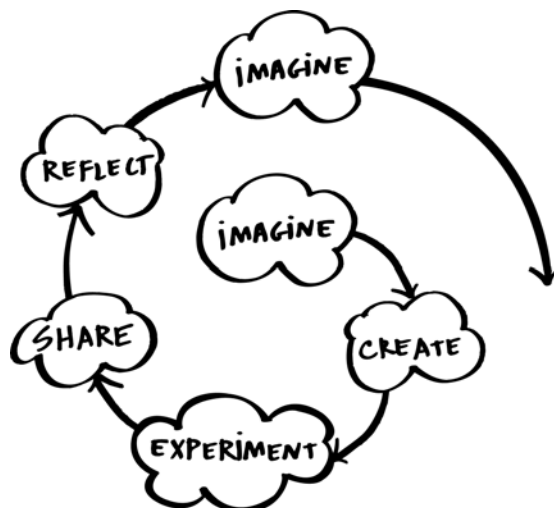
Creating projects in Scratch also helps students develop a deeper level of **fluency** with digital technology. What do we mean by fluency? To be considered fluent in English, Spanish, or other language, you must learn not only how to read but also to write – that is, how to express yourself with the language. Similarly, to be fluent with digital technology, you must learn not only how to interact with the computer but also to create with it.

Of course, most students will not grow up to become professional programmers, just as most will not become professional writers. But **learning to program** offers benefits for everyone: it enables students to express themselves more fully and creatively, helps them develop as logical thinkers, and helps them understand the workings of the new technologies that they encounter everywhere in their everyday lives.

References
Rethinking Learning in the Digital Age
http://www.media.mit.edu/~mres/papers/wef.pdf

Learning for the 21st Century (http://www.21stcenturyskills.org/)

Lifelong Kindergarten Group, MIT Media Lab

SCRATCH